

### LIST OF THE CLAIMS

Please amend Claims 1, 12, 17, 25, 30 and 33 as follows:

1. (Currently Amended) A plug-in to a host, the plug-in providing one or more special capabilities to the host, the plug-in comprising:

core means for conducting typical operation of the plug-in by which the one or more special capabilities are carried out; and

interface means for interfacing between the core means and a calling entity with respect to operational status of the plug-in, wherein the operational status is provided to a second plug-in that requested the operational status from the plug-in.

2. (Original) The plug-in of claim 1, wherein:

the interface means is a first interface means; and

the plug-in further comprises

second interface means for interfacing between the core means and the host with respect to the typical operation.

3. (Original) The plug-in of claim 1, wherein:

the host tends one or more storage devices and represents a part of a storage area network (SAN); and

the calling entity represents a storage manager of the SAN.

4. (Original) The plug-in of claim 1, wherein, relative to a client-server architecture, the following relationships apply:

the calling entity represents a client relative to the plug-in; and

the plug-in represents a server relative to the calling entity.

5. (Original) The plug-in of claim 4, wherein the client-server architecture is the JCore architecture such that the calling entity is a JCore client plug-in and the plug-in is a JCore server plug-in.

6. (Original) The plug-in of claim 1, further comprising:  
butler means for gathering operational status information (Op\_Stat\_Info)  
representing the operational status of the plug-in.

7. (Original) The plug-in of claim 6, wherein the butler means is operable for  
performing the gathering of the Op\_Stat\_Info in an on-going manner while the plug-in is  
plugged-into the host.

8. (Original) The plug-in of claim 7, wherein the butler means is further  
operable to  
cause one or more pieces of the Op\_Stat\_Info to be stored upon the one or more  
pieces being gathered initially, respectively; and  
update the one or more pieces, respectively, as is appropriate relative to the on-  
going manner by which the butler means gathers the Op\_Stat\_Info.

9. (Original) The plug-in of claim 8, further comprising:  
status data object (DO) means for storing the Op\_Stat\_Info;  
wherein  
the butler means is further operable for causing the Op\_Stat\_Info to be  
stored via the status DO means,  
the status DO means is accessible by the calling entity via the interface  
means.

10. (Original) The plug-in of claim 6, wherein the butler means is operable for  
initiating the gathering of Op\_Stat\_Info in response to receiving a request from  
the calling entity, and  
for performing the gathering, once initiated, of the Op\_Stat\_Info in an ad hoc  
manner.

11. (Original) The plug-in of claim 10, further comprising:

status data object (DO) means for storing the Op\_Stat\_Info;  
wherein the butler means is further operable for  
causing the Op\_Stat\_Info to be stored via the status DO means, and  
passing, upon completion of the gathering, the status DO to the calling  
entity.

12. (Original) A first plug-in to a host, the first plug-in providing one or more special capabilities to a host, the first plug-in comprising:  
core means for conducting typical operation of the first plug-in by which the one or more special capabilities are carried out; and  
interface means for interfacing between the core means and one or more second plug-ins loaded on one or more external, relative to the host, entities, respectively, regarding respective operation status of the one or more second plug-ins.

13. (Original) The first plug-in of claim 12, wherein:  
the interface means is a first interface means; and  
the first plug-in further comprises  
second interface means for interfacing between the core means and the host with respect to the typical operation.

14. (Original) The first plug-in of claim 12, wherein:  
the host is a first host;  
the one or more external entities are one or more second hosts that tend one or more storage units, respectively;  
the first host and the one or more second hosts represent a part of a storage area network (SAN); and  
the host represents a storage manager of the SAN.

15. (Original) The first plug-in of claim 14, wherein, relative to a client-server architecture, the following relationships apply:  
the first plug-in represents a client relative to the second plug-in; and

the second plug-in represents a server relative to the first plug-in.

16. (Original) The first plug-in of claim 15, wherein the client-server architecture is the JCore architecture such that the first plug-in is a JCore client plug-in and the second plug-in is a JCore server plug-in.

17. (Currently Amended) A method of operating a plug-in to a host, the plug-in providing one or more special capabilities to the host, the method comprising:  
interfacing to make available, from the plug-in to an external calling entity relative to the host, operational status information (Op\_Stat\_Info) regarding the plug-in, wherein the operational status is provided to a second plug-in that requested the operational status from the plug-in.

18. (Original) The method of claim 17, further comprising:  
exchanging typical information between the plug-in and the host, the exchange of the typical information being a part of typical operation of the plug-in by which the one or more special capabilities are carried out.

19. (Original) The method of claim 17, wherein:  
the host tends one or more storage devices and represents a part of a storage area network (SAN); and  
the Op\_Stat\_Info is made available to a storage manager of the SAN.

20. (Original) The method of claim 17, further comprising:  
gathering the Op\_Stat\_Info in an on-going manner while the plug-in is plugged into the host.

21. (Original) The method of claim 20, further comprising:  
storing one or more pieces of the Op\_Stat\_Info upon initially gathering the one or more pieces, respectively; and

updating the one or more pieces, respectively, as is appropriate relative to the on-going manner of the gathering step.

22. (Original) The method of claim 21, further comprising:  
using a status data object (DO) to store the Op\_Stat\_Info; and  
making the status DO accessible by the calling entity.

23. (Original) The method of claim 20, further comprising:  
receiving a request from the calling entity for the Op\_Stat\_Info;  
initiating the gathering step in response to receiving the request; and  
performing the gathering step, once initiated, in an ad hoc manner.

24. (Original) The method of claim 23, further comprising:  
using a status data object (DO) to store the Op\_Stat\_Info; and  
passing, upon completing the gathering step, the status DO to the calling entity.

25. (Original) A machine-readable medium including instructions execution of which by a host produces a first plug-in, the first plug-in providing one or more special capabilities to the host, the machine-readable instructions comprising:

a core code segment for conducting typical operation of the first plug-in by which the one or more special capabilities are carried out; and

an interface code segment for interfacing between the core code segment and one or more second plug-ins loaded on one or more external, relative to the host, entities, respectively, regarding respective operational status of the one or more second plug-ins.

26. (Original) The machine-readable instructions of claim 25, wherein:  
the interface code segment is a first interface code segment; and  
the machine-readable instructions further comprise  
a second interface code segment for interfacing between the core code segment and the host with respect to the typical operation.

27. (Original) The machine-readable instructions of claim 25, wherein:  
the host is a first host; and  
the one or more external entities are one or more second hosts that tend one or more storage units, respectively;  
the first host and the one or more second hosts represent a part of a storage area network (SAN); and  
the first host represents a storage manager of the SAN.

28. (Original) The machine-readable instructions of claim 25, wherein, relative to a client-server architecture, the following relationships apply:  
the first plug-in represents a client relative to the second plug-in; and  
the second plug-in represents a server relative to the first plug-in.

29. (Original) The machine-readable instructions of claim 28, wherein the client-server architecture is the JCore architecture such that the first plug-in is a JCore client plug-in and the second plug-in is a JCore server plug in.

30. (Original) A method of operating a first plug-in to a host, the plug-in providing one or more special capabilities to a first host, the method comprising:  
obtaining operational status information (Op\_Stat\_Info) from one or more second plug-ins loaded on one or more external, relative to the host, entities, respectively.

31. (Original) The method of claim 30, further comprising;  
exchanging typical information between the first plug-in and the host, the exchange of the typical information being a part of typical operation of the plug-in by which the one or more special capabilities are carried out.

32. (Original) The method of claim 30, wherein:  
the host is a first host; and

the one or more external entities are one or more second hosts that tend one or more storage units, respectively;

the first host and the one or more second hosts represent a part of a storage area network (SAN); and

the first host represents a storage manager of the SAN.

33. (Currently Amended) A machine-readable medium including instructions execution of which by a host produces a plug-in, the plug-in providing one or more special capabilities to the host, the machine readable instructions comprising:

a core code segment execution of which causes the one or more special capabilities to be carried out during typical operation of the plug-in; and

a status interface code segment for interfacing between the core and a calling entity and by which operational status of the plug-in is made available to the calling entity, wherein the operational status is provided to a second plug-in that requested the operational status from the plug-in.

34. (Original) The machine-readable instructions of claim 33, further comprising:

a central interface code segment for interfacing between the core portion and the host and by which typical information is exchanged between the plug-in and the host, the exchange of the typical information being a part of the typical operation.

35. (Original) The machine-readable instructions of claim 33, wherein:

the host tends one or more storage devices and represents a part of a storage area network (SAN); and

the calling entity represents a storage manager of the SAN.

36. (Original) The machine-readable instructions of claim 33, wherein the plug-in, the host and the calling entity are configured according to the JCore architecture such that the client entity represents a client relative to the plug-in and the plug-in represents a server relative to the calling entity.